

Memòria del TFC

Gestió dels arxius duplicats distribuïts

Joan Antoja Sabin (joans@uoc.edu)

Treball de Final de Carrera

Enginyeria Tècnica en Informàtica de Sistemes

Juny del 2011

Aquest treball està subjecte a una llicència [Creative Commons By-Nc-Nd](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Índex de continguts

1. Introducció.....	3
1.1. Contextualització.....	3
1.2. Justificació.....	3
1.3. Objectius.....	4
2. Detecció de necessitats.....	5
2.1. Casos reals.....	5
2.2. Abstracció.....	7
3. Espai de treball.....	9
3.1. Ítems a valorar.....	9
3.2. Proves i procediment.....	10
3.3. Criteris per cercar les aplicacions a provar.....	11
3.4. Entorn de treball.....	11
3.5. Simulant naturalitat.....	13
4. Treball de camp i recollida de dades.....	14
4.1. Llistat d'aplicacions.....	14
4.2. Consideracions tècniques i de nomenclatura.....	16
4.3. Programaris aptes.....	22
4.4. Programaris descartats.....	46
5. Conclusions.....	47
5.1. Quines característiques hauria de tenir un programari de sincronització?.....	47
5.2. Proposta.....	48
5.3. Propostes de millora	49
5.4. Recomanacions.....	50
6. Annexes.....	51
6.1. Modificació automàtica d'arxius.....	51
6.2. Comprovació automàtica dels canvis.....	54
7. Bibliografia.....	55

1. Introducció

1.1. Contextualització

Amb l'arribada de les xarxes a tots els àmbits (personal, laboral...), i en especial Internet, així com el desenvolupament de les eines de treball en línia, que permeten la ubiqüitat, s'ha fet que es disposi de molta informació i que es treballi amb molta informació.

Els arxius que emmagatzemen aquesta informació amb la que es treballa, sovint estan distribuïts seguint la nostra estela de moviment, i desada físicament en diferents suports i ubicacions: discs en local, discs externs, el núvol...

Per altra banda, la diversificació de sistemes operatius, així com maquinari (ordinadors amb arquitectura Intel, Mac..., smartphones...) fa que l'accés a la informació no sigui des de les mateixes plataformes.

L'ús dels sistemes basats en GNU/Linux en general (i del programari lliure en concret) s'ha estès, tant com a entorn d'escriptori així com a plataformes o servidors massius degut a la llicència que li permet la còpia, millora i distribució, com l'eficiència contrastada d'aquests programaris i l'ús de formats oberts.

El programari lliure, doncs, facilita la portabilitat. Per tant, serveix com a base de treball per trobar una solució el màxim extensible possible.

1.2. Justificació

El treball diari en diferents entorns, tant pel que fa a sistemes operatius, com maquinari com llocs físics diferents fa que la informació hagi d'estar a tot arreu. Aquesta informació ha d'estar actualitzada. A més a més, hi ha la possibilitat de l'ús de les diferents possibilitats d'emmagatzematge (local, disc extern, núvol...).

Tot i que les eines de treball al núvol (per exemple, Google Docs, EyeOS...) són molt completes, el fet d'utilitzar els programaris instal·lats als ordinadors encara segueix sent necessari, per les seves capacitats i rapidesa. Més quan, estem treballant amb grans volums d'informació (com grans documents, imatges, audio...).

Per altra banda, en molts casos, les solucions ofertes per aquesta gestió, estan orientades a públics molt concrets (per exemple desenvolupadors amb el control de versió), o a

entorns empresarials en els que es tenen milers de treballadors (per exemple els programaris de treball en grup).

Un fet que hi ajuda pot ser no disposar d'una bona connexió a Internet tenint en compte les necessitats d'anar desant la informació, o no prou ràpida per fer uns accessos pràctics o poder seguir treballant quan hi ha mobilitat que pot generar talls en la connexió.

A més a més, el fet de tenir còpies de diferents arxius, requereix un esforç de gestió, que creix desproporcionadament en la mesura que van creixent els espais i els arxius.

1.3. Objectius

- Conèixer i avaluar les eines que permeten mantenir varies còpies dels arxius.
- Crear una proposta a partir dels resultats de l'avaluació.

2. Detecció de necessitats

Per poder fer una bona valoració de les aplicacions que realitzen sincronitzacions, primer de tot caldrà establir què és el que cal valorar de cadascuna, així com que cal tenir en compte, per després poder extreure unes conclusions i fer una proposta.

2.1. Casos reals

El treball partirà de quatre casos reals en els quals, per diferents motius, esdevé la necessitat de mantenir diverses còpies de diferents arxius. La informació s'extreu a partir d'entrevistes amb les persones (o responsables d'informàtica) de cada cas. A part de conèixer l'activitat general de cadascuna, es recull principalment la següent informació:

- Dedicació de l'empresa.
- Recursos humans.
- Ubicació física del personal i de la infraestructura.
- Volum clients o usuaris amb als que han de tractar.
- Volum d'informació utilitzada en la seva activitat habitual.
- Volum d'informació modificada/creada per període de temps.
- Tipus d'informació amb la que treballen.
- Maquinari i sistemes operatius, és a dir, infraestructura tecnològica de la que disposen.
- Problemàtica amb la que es troben.

a) Cas número 1

L'entitat F es dedica a l'educació, la conciliació de vida laboral i familiar i la dinamització comunitària. Té 8 treballadors, dels quals 4 estan físicament ubicats a on es realitzen els projectes, que són equipaments i escoles. Cada any participen directament de les activitats organitzades per F, unes 400 persones. Tot això es tradueix en uns 50 documents creats o modificats setmanalment. El tipus d'informació amb el que treballa són documents de text, fulls de càlcul i imatges. Tots els ordinadors de l'entitat són bàsics,

és a dir, pensats per treball d'oficina i amb sistema operatiu Ubuntu, però els dels equipaments poden tenir altres sistemes operatius (Windows o Mac). Tot i tenir connexió remota al servidor de l'entitat, el fet de connectar-s'hi des de connexions lentes o sobreutilitzades, fa que sovint l'accés a un arxiu model de, per exemple, autorització de participació d'un menor d'un centenar de kilobytes a una activitat, suposi minuts.

b) Cas número 2

A és un autònom que es dedica a la programació web i al manteniment de sistemes. Normalment treballa des del seu despatx, tot i que el que produeix normalment està ubicat en servidors remots; per altra banda, aprofita els desplaçaments o les esperes per avançar feina. Té tres clients fixos i d'ocasionals. Cadascun dels projectes web que gestiona pot tenir uns 20.000 arxius. Una actualització de la base de treball canvia uns 500 arxius, més les modificacions que demana el client, que són d'uns 20 arxius. Els tipus d'arxius són entre codi HTML, imatges i fulles d'estil propis, així com el que aporta el CRM base o gestor que utilitzi. Ell utilitza en les seves màquines amb sistema operatiu Kubuntu, i tots els servidors disposen de RedHat/CentOS. Gestionar una còpia exacta per treballar en local de la versió en producció i la versió en desenvolupament és molt costós en quan a temps, tenint en compte que no és l'únic que introdueix canvis, i que els hostings no preveuen control de versions.

c) Cas número 3

C és autònoma i es dedica al disseny gràfic. Treballa sola, tot i que en temporada alta contracta una o dues persones per ajudar-la a treure feina. La feina grossa es fa des de la oficina, tot i que mínim dos cops per setmana visita clients, als quals els hi fa propostes, demostracions, entregues... Cada projecte pot tenir, entre grafisme base, proves intermèdies i acabats uns 3 gigabytes d'informació (més si es tracta de vídeos). Un projecte, que dura entre 2 i 6 mesos, comença amb un document d'uns pocs kilobytes i pot acabar en uns 20 gigabytes. La informació emmagatzemada està composta majoritàriament per imatges i vídeos. Utilitza un Mac portàtil i varis discs durs externs per emmagatzemar la informació. La problemàtica amb que es troba és que al treballar amb l'accés a la informació en local, és a dir, en el disc dur és molt més ràpida. Per altra banda, no és pràctic ni segur viatjar amb els discs durs externs, però per un tema de volum tampoc pot tenir tota la informació a l'ordinador de treball.

d) Cas número 4

L'empresa B es dedica a oferir serveis d'emmagatzematge, tan en servidors externs propis de B com en els que els mateixos clients tenen in-situ. L'empresa té 10 treballadors dedicats al servei de manteniment de les màquines, que treballen remotament sempre que és possible, però que es desplacen, si és necessari. L'empresa té uns 200 clients repartits pel país. Cada persona de l'equip tècnic disposa d'un ordinador portàtil en el que ha de portar tota la documentació de configuració dels clients, junt amb els programaris propis, de suport, de control de maquinari... que fa que passi lleugerament els 10 gigabytes. Cada cop que hi ha una actualització de programari, tot l'equip tècnic l'ha de tenir, així com a cada resolució d'incidència s'ha de generar un informe, que de mitjana són unes 30 a la setmana. A part dels programaris, hi ha documents de text i esquemes. Tots els portàtils tenen arranc dual amb Windows XP i Ubuntu. La problemàtica és que tota la informació ha d'estar al portàtil, ja que poden trobar-se amb incidències, per exemple, de connectivitat; per altra banda, tot ha d'estar actualitzat a l'última versió, ja sigui, per exemple, per no instal·lar un programari del que ja se'n coneixen errades de seguretat, o de conèixer amb exactitud quina és la configuració de xarxa.

2.2. Abstracció

L'abstracció feta a partir del recollit en les entrevistes en surt que:

- han de mantenir diverses còpies exactes de la informació,
- la informació s'ha de desplaçar,
- la limitació està en el volum d'informació respecte la connexió o els dispositius d'emmagatzematge,
- la complexitat i el temps que comporta fer-ho manualment,
- es treballa amb informació confidencial, i en alguns casos sensible.
- tot i que el treball es realitza en local, és necessari que tant bon punt sigui possible, aquesta informació estigui disponible per tothom,
- la informació és manipulada per diferents persones,
- tot acaba passant per un punt central,

- dins de cada entorn la informació s'emmagatzema de maneres diferents,
- en cada entorn hi ha sistemes operatius diferents, i
- que hi ha canvis en l'organització de la informació.

I també s'ha de tenir en compte que només s'ha fet una detecció de necessitats per conèixer en quins problemes es troben, però per proposar una nova solució, a part dels ítems mencionats anteriorment s'han de tenir en compte també:

- quina infraestructura necessita cada solució,
- quins canvis s'haurien de fer,
- sobre quina versió de cada programari estem treballant,
- que utilitza cada programari, i
- que implica o quina millora suposa afegir la solució.

A més, hi ha una part, que en principi queda fora de l'abast del treball, que és que hi ha altres sistemes operatius, més enllà del GNU/Linux. En aquest aspecte, la proposta és treballar la part de portabilitat que normalment és inherent al programari de codi lliure.

Si més no, també es tindrà en compte com es resoldria aquesta situació en un entorn de programari lliure

3. Espai de treball

3.1. Ítems a valorar

A partir dels dos punts anteriors, és a dir, de la informació dels casos, junt amb l'abstracció i els punts afegits de cara a plantejar una possible solució sorgeixen els següents ítems a tenir en compte de cada aplicació valorada:

1. automatització, capacitat de treballar sense (o amb la mínima) intervenció de l'usuari,
2. paràmetres que controla, o com detecta que un arxiu ha canviat, en que es basa per fer una sincronització,
3. estructura, o quin model de xarxa utilitza, per exemple, client/servidor, client, com emmagatzema, topologia que utilitza...,
4. compressió de la informació quan es transmet,
5. seguretat en la transmissió,
6. detecció de conflictes,
7. integritat de les dades,
8. emmagatzematge suportat,
9. portabilitat, en quins altres sistemes operatius està (o pot estar) disponible,
10. llicència,
11. protocol utilitzat per transmetre,
12. versió del programari utilitzada per fer les proves,
13. com gestiona el reanomenar d'arxius,
14. com gestiona el moviment d'arxius,
15. direccionalitat del programari, és a dir, en quina direcció treballa,
16. infraestructura necessària, o quines dependències té,
17. cost econòmic,
18. cost humà i temporal d'utilització/instal·lació,

19. interfície, i

20. velocitat de transferència.

3.2. Proves i procediment

Per cada aplicació que es provarà el procediment serà el següent:

Primer de tot, instal·lar l'aplicació, en les màquines virtuals que faci falta, és a dir, pot ser que s'hagin de fer intervencions en les tres, en dues o en una. Això es farà a través de la consola de cada màquina virtual amb la comanda:

```
$ sudo apt-get install <app>
```

Un cop l'aplicació estigui correctament instal·lada, juntament amb totes les seves dependències, el següent pas serà, si és necessari, configurar-la.

Un cop fet aquest pas s'omplirà la fitxa corresponent amb la informació teòrica sobre l'aplicació (versió, dependències, llicència...) i seguidament començaran les proves en el següent ordre:

1. Primera còpia
2. Modificació d'arxius en un costat
3. Modificació d'arxius diferents en dos costats
4. Modificació dels mateixos arxiu en els dos costats
5. Moure arxius/directoris en un directori
6. Reanomenar arxius/directoris

que de manera genèrica cobreixen les situacions habituals i possibles. Durant i després de cada prova, es repassarà com han quedat els arxius en cada costat. Per veure per on s'han propagat les diferències, hi haurà els arxius amb els arxius modificats, i un *script* per comparar en ambdues quins arxius són iguals i quins diferents.

De la mateixa manera, mentre es facin les proves es recollirà la informació necessària dels ítems a valorar, des d'un punt de vista pràctic.

3.3. Criteris per cercar les aplicacions a provar

En aquest punt es tracta, dins el munt d'aplicacions que existeixen actualment, fer la tria de les que seran susceptibles de ser provades. S'utilitzaran dues vies de cerca, una serà el repositori de la distribució Ubuntu i l'altre el cercador Google. D'aquesta manera, no només hi ha el programari inclòs en la distribució, i dóna peu a fer un contrapunt.

Les paraules clau que s'utilitzaran per fer aquesta cerca seran “*sync*”, “*files*”, “*synchronize*”, “*synchronization*” i “*directories*”, combinant-les perquè sempre hi aparegui el concepte de sincronització. En el cas del cercador, al no ser especialitzat en programari, se li afegirà la paraula “*software*” o “*application*”. Aquí s'hauran de discriminar els resultats que no estiguin disponibles per GNU/Linux.

En el cas de la cerca en el repositori de la distribució Ubuntu s'utilitzarà la comanda següent:

```
$ apt-cache search <paraula/es>
```

A partir dels resultats, s'elaborarà una llista amb els programaris a provar, juntament amb una breu descripció. En aquesta llista, no s'hi inclouran els que, a primera vista a jutjar per la descripció, no compleixen, ni s'acosten a la funció de la sincronització, així com els que són molt semblants a d'altres que ja es provaran.

3.4. Entorn de treball

Per poder fer les proves d'una manera neutral, s'utilitzaran màquines virtuals amb un joc d'arxius de diferents tipus i mides. D'aquesta manera, no s'afectarà el funcionament de cap entorn de treball en producció.

De màquines virtuals se'n crearan 3 amb les característiques per simular els escenaris següents:

- Màquina virtual 1 (M1): màquina de la persona que treballa suposadament en una de les empreses. Aquesta tindrà els programaris de sincronització instal·lats, i és des de la qual es mesuraran les proves. Tindrà connexió a una xarxa virtual que l'enllaçarà amb altres màquines virtuals. També disposarà d'una simulació de disc dur extern USB, per fer sincronitzacions en local. I òbviament una còpia de tots els arxius.

- Màquina virtual 2 (M2): igual que la màquina virtual 1, però només amb l'objectiu de simular l'ús de dades per una altra persona.
- Màquina virtual 3 (M3): farà de servidor, o de punt de trobada per sincronitzar en cas que sigui necessari.

El sistema hoste és ordinador amb processador Intel Core 2 6300 a 1,86GHz amb 6GB de RAM i 160 GB de disc dur. El programari utilitzat per la virtualització serà VMware Player 3.0.1 corrent sobre un sistema operatiu Kubuntu 10.04.

Els sistemes clients seran simulacions d'arquitectura Intel, amb un processador, 512MB de RAM i 40GB de disc dur. El sistema operatiu instal·lat serà Ubuntu 10.10 i386. A més, hi haurà un disc dur virtual que connectarà per la interfície USB.

Les màquines virtuals tindran arquitectura i386, amb un sol processador, per poder fer una simulació que sigui el màxim compatible amb les situacions descrites.

La creació de les màquines virtuals es fa utilitzant la utilitat "Easy Install", de manera que només facilitant el nom d'usuari i la contrasenya, es fa una instal·lació bàsica del sistema operatiu.

Per enllaçar les tres màquines virtuals, la opció de configuració de xarxa és "*bridged network*", és a dir, que les targetes de xarxa virtuals simulin ser unes connectades a la de la màquina hoste. D'aquesta manera passen a ser 3 màquines més de la xarxa.

La configuració de la xarxa serà la següent:

- M1: 192.168.0.231
- M2: 192.168.0.232
- M3: 192.168.0.233

de manera que les tres màquines seran visibles entre elles, actuant com si estiguessin en una mateixa xarxa local. D'aquesta manera s'equiparen a totes les situacions descrites, és a dir, connectades a través de la xarxa.

Un cop posats en marxa els sistemes operatius, s'instal·laran les actualitzacions de seguretat i de correcció d'errors per evitar fallades per problemes ja solucionats.

Per tal de poder compartir arxius de manera ràpida entre les màquines virtuals, i alhora

poder poder fer proves en xarxa amb els programaris que no contemplen de manera nativa aquesta opció, hi haurà un directori compartit de la màquina M3. El sistema que es configurarà serà *NFS*, de manera per les màquines M1 i M2 es muntarà (*/mnt/shared/*) com un directori més del sistema de fitxers.

3.5. *Simulant naturalitat*

Per tal de crear un ambient de treball en que s'acosti a un de real, les proves es faran amb un directori que contindrà uns 1000 arxius de mides entre 1KB i 4GB, i de documents de text fins a pel·lícules.

Per fer les modificacions, es generaran uns senzills *scripts* encarregats de fer-ho, de manera que només en modifiquin una part determinada, introduint-lis nou contingut.

És important aquest pas, per donar-li un punt d'aleatori, en el sentit que no es modifiquin sempre els mateixos arxius. Per altra banda, també canviar-lis el contingut, perquè segons quin programari, per exemple, potser no contrastarà només el bit de modificat o la data, sinó el contingut.

A través del directori compartit entre les màquines es compartirà la informació per poder comparar i contrastar com ha anat cada còpia, així com per saber quins arxius s'han de modificar en el cas que es vulgui modificar el mateix arxiu en ambdues bandes.

4. Treball de camp i recollida de dades

Un cop plantejat el problema comença la posada en pràctica. En els següents apartats s'aniran desgranant punt per punt les diferents etapes. Val a dir, que ha calgut especificar alguns conceptes plantejats per no caure en dubtes, així com fer alguns canvis o obviar algunes parts per les circumstàncies donades.

Primer hi ha el llistat d'aplicacions que entren en el joc de proves, després un conjunt de consideracions tècniques i de nomenclatura que han sorgit al fer les proves, així com per unificar criteris i acotar el problema, i finalment, les fitxes dels programaris que poden donar la solució al nostre problema o que aporten algun concepte/idea que pot ajudar a enfocar. Al final de tot, els programaris descartats, perquè, o queden fora de l'abast dels objectius o perquè són molt semblants a d'altres ja provats.

4.1. Llistat d'aplicacions

Un cop fet un primer triatge a partir de la descripció, queda de la següent manera, tenint en compte que hi ha aplicacions susceptibles de ser provades i d'altres que queden descartades, però de les que se'n poden treure idees.

Pel que fa a la cerca en el repositori d'Ubuntu, es partirà de:

Programari	Breu explicació que l'acompanya
amanda-server	Advanced Maryland Automatic Network Disk Archiver (Server)
amanda-client	Advanced Maryland Automatic Network Disk Archiver (Client)
backintime-kde	KDE frontend for backintime
backintime-gnome	GNOME frontend for backintime
backintime-common	simple backup/snapshot system
backup-manager	command-line backup tool
backuppc	high-performance, enterprise-grade system for backing up PCs
conduit	<i>synchronization tool for GNOME</i>
csync2	<i>cluster synchronization tool</i>
dirvish	<i>Filesystem based backup system using rsync</i>
duplicity	<i>encrypted bandwidth-efficient backup</i>
flamethrower	<i>Multicast file distribution utility</i>
flickrfs	<i>virtual filesystem for flickr online photosharing service</i>

Programari	Breu explicació que l'acompanya
gerstensaft	<i>Frontend for Simple Asynchronous File Transfer</i>
grsync	<i>Grsync is a rsync GUI (Graphical User Interface).</i>
inosync	<i>notification-based directory synchronization daemon</i>
komparator	<i>directories comparator for KDE</i>
krusader	<i>twin-panel (commander-style) file manager</i>
mirror	<i>keeps FTP archives up-to-date</i>
opensync-plugin-file	<i>Opensync file plugin</i>
rdiff-backup	<i>remote incremental backup</i>
rdiff	<i>Binary diff tool for signature-based differences</i>
rdup	<i>utility to create a file list suitable for making backups</i>
rsnapshot	<i>local and remote filesystem snapshot utility</i>
rsync	<i>fast remote file copy program (like rcp)</i>
sendfile	<i>Simple Asynchronous File Transfer</i>
simba	<i>next generation mirroring tool</i>
subversion-tools	<i>Assorted tools related to Subversion</i>
syrep	<i>A generic file repository synchronization tool</i>
ubuntuone-client-tools	<i>Ubuntu One client tools</i>
unison-gtk	<i>A file-synchronization tool for Unix and Windows with GTK+ interface</i>
unison	<i>A file-synchronization tool for Unix and Windows</i>
zsync	<i>client-side implementation of the rsync algorithm</i>

Pel que fa a la cerca mitjançant Google:

Programari	Breu descripció que l'acompanya
Dropbox	<i>Any file you save to Dropbox also instantly saves to your computers, phones, and the Dropbox website.</i>
Egnyte	<i>The Egnyte Cloud File Server is an easy-to-use online storage solution for storing business files of any size or type online in secure, permission-based folders</i>
Wuala	<i>Backup. Sync. Share. Access Everywhere. All data encrypted on your computer.</i>
Zumodrive	<i>ZumoDrive is the hybrid cloud storage for all of your media.</i>

4.2. Consideracions tècniques i de nomenclatura

Primerament un apunt de nomenclatura, quan es parla de “l'altre costat” és una referència a la còpia que hi ha en la màquina remota amb la qual s'ha sincronitzat. Quan es parla del més “nou” és una referència a l'arxiu que té una data més jove.

Com que s'estan valorant aplicacions amb característiques semblants (sinó iguals), amb l'objectiu de que la fitxa de cada programari provat no s'estengués innecessàriament, així com per unificar conceptes i clarificar que significa cada resposta, a continuació es presenta una llista, que, per cada element llista les opcions més comunes amb una explicació.

Abans, però, per acotar el problema, i centrar-se en els objectius del treball:

La velocitat de transferència ha quedat fora de les fitxes de cada programari provat, doncs en les condicions creades, no és mesurable amb fiabilitat. Això és així perquè, per un costat, la connexió no passa per un medi físic de xarxa, doncs les màquines virtuals estan a la mateixa màquina. A més, totes les màquines virtuals llegeixen i escriuen en el mateix disc dur físic, al que se li sumen les operacions que fa el gestor de màquines virtuals per dimensionar l'espai utilitzat segons necessitats.

Per tant, des de les primeres proves es va veure que les lectures de velocitat no eren realistes, i en una mateixa prova repetida varies vegades podien canviar de l'ordre de 10 vegades.

Integritat de les dades també queda fora, doncs no s'ha donat ni una sola corrupció de dades. Això és així, en part per les bones condicions de l'entorn (màquines virtuals en la mateixa màquina amb un enllaç en perfectes condicions), tant pel fet que majoritàriament els protocols utilitzats estan ja dissenyats per garantir la integritat. Per tant, no hi ha manera senzilla en aquest entorn de provocar errades.

a) Llicència:

- Artistic License: veure Open Source Initiative OSI - The Artistic License
- GPL: veure GNU General Public License.
- GPL (client): només una part del programari que entra en joc, en aquest cas el client, disposa d'aquesta llicència.

b) Automatització:

- Si: el programari inclou mecanismes propis per funcionar de manera autònoma, o dit d'una altra manera, sense intervenció de l'usuari.
- No directa: es pot automatitzar, doncs el programari pot funcionar sense interacció de l'usuari, però no està previst en la funcionalitat pròpia del mateix, és a dir, que ha de ser cridat per un altre procés, com per exemple, un *cron*.
- No: el programa no es pot automatitzar de cap manera.

c) Portabilitat

Entenent que partim a partir de GNU/Linux:

- Android / iPhone / Blackberry: portat als sistemes operatius per *smartphones*.
- Gtk: depèn d'aquestes llibreries, és a dir, que tot i que teòricament serà portable, això està condicionat a que les llibreries estiguin disponibles en altres sistemes operatius.
- KDE: igual que en el cas anterior, però amb aquestes llibreries.
- Mac: portat al sistema operatiu d'Apple. Es considera les versions 10 i superiors.
- Windows: portat al sistema operatiu de Microsoft, ja sigui de manera nativa o mitjançant el projecte *cygwin*. Es considera les versions XP i superiors.

d) Dependències

Pel que fa a les dependències, només es comenten els programaris (o paquets) més generals, sense entrar en mòduls concrets o pressuposant que el sistema operatiu base està instal·lat.

- Apache: servidor web.
- Perl: programat amb aquest llenguatge, requereix de l'interpret de codi i en funció de les crides que utilitzi, llibreries de la API del mateix.
- Samba: implementació lliure del sistema de compartir arxius i impressores de Microsoft Windows.
- Bzip2: compressor de codi lliure.

- Tar: empaquetador de codi lliure.
- Gnupg: implementació lliure del sistema de criptografia PGP.
- Python: programat amb aquest llenguatge, requereix de l'interpret de codi i en funció de les crides que utilitzi, llibreries de la *API* del mateix.
- Gnome: requereix de les llibreries d'aquest entorn gràfic per funcionar.
- Nautilus: requereix del programari de gestió d'arxius per defecte de l'entorn gràfic Gnome.

e) Interfície:

- Gràfica: el programari només pot interaccionar amb l'usuari a través de l'entorn gràfic.
- Consola: el programari només pot interaccionar amb l'usuari a través de la consola.
- Gràfica/Consola: el programari pot interaccionar amb l'usuari tant a través de l'entorn gràfic com a través de la consola.
- Web: el programari es pot gestionar a través d'un navegador independentment del sistema operatiu.

f) Estructura:

- Client: només és necessari el programari que s'està utilitzant per funcionar.
- Client – Client: el programari es connecta a les altres màquines a través dels altres clients (que al seu torn fan de servidors), i viceversa. És el sistema que es coneix com a *peer-to-peer*.
- Client – Servidor: el programari necessita d'un programari intermedi servidor, el qual només fa aquesta funció.

g) Emmagatzematge suportat

Pel que fa a l'emmagatzematge queden fora de l'abast d'aquest treball els mitjans que no permeten escriptura/lectura àgil, o que no s'utilitzen amb el propòsit de treball diari. Com exemple concret hi hauria mitjans tipus CD/DVD i cintes.

També en aquest sentit, cal matisar que es refereix a dispositius físics, independentment de qui ho gestioni, sobretot enfocant en l'àmbit local. En aquest sentit, sobretot pel que fa als remots subcontractats, com per exemple Amazon S3, no hi ha manera de saber en quin dispositiu està desada físicament la informació.

A partir d'aquí, les opcions disponibles són:

- Qualsevol: s'entén que ja sigui directament o per interacció a través del sistema operatiu (local o remot) pot desar informació en qualsevol tipus de suport.
- Qualsevol (amb enllaços forts): es refereix a sistemes de fitxers que tenen la capacitat de vincular un arxiu a varis noms, és a dir, que des de diferents camins en realitat s'estigui consultant la mateixa informació. La conseqüència d'això és que només hi hagi una sola còpia de l'arxiu escrita físicament al disc.

h) Protocol de transmissió

Quin protocol (o concepte) s'utilitzarà per fer les transferències.

- Delta encoding: concepte de transmissió en el qual, que al sincronitzar dos arxius, només s'envien els canvis que s'han donat.
- Gnome: s'utilitza la infraestructura d'aquest entorn gràfic per realitzar les operacions de còpia.
- Kioslave: igual que en el cas anterior, però amb l'entorn gràfic KDE.
- Rsync: al treballar en remot, necessita que el mateix programari estigui instal·lat a l'altre costat, ja sigui, escoltant en mode servidor o només el client amb la possibilitat d'executar-lo des d'una crida remota, per exemple a partir d'una connexió via SSH.
- Sistema: utilitza les crides al sistema per realitzar aquesta operació, per exemple, la comanda *cp*.
- u1sync: sistema de transmissió creat pel sistema Ubuntu One.
- FTP: File Transfer Protocol.
- Amazon S3: sistema de transmissió amb el servei S3 de l'empresa Amazon.

- SSH: Secure Socket Layer, com a tal no té la funció de copiar, sinó que s'utilitza com a embolcall per d'altres protocols, com per exemple el *rsync* que no xifra les dades.
- Samba: transmissió a través del protocol de compartir arxius de Windows.
- NFS: Network File System.
- RSH: Remote SHell.

i) Compressió en la transmissió:

- bzip2: als arxius, abans de ser enviats se'ls aplica un procés de compressió perquè ocupin menys, per tant, no és en la transmissió, sinó en un estadi previ, tot i que el resultat és que el volum transmès és menor.
- Delta encoding / *rsync*: no es tracta de compressió pròpiament, doncs la informació no és tractada durant l'enviament perquè ocupi menys, sinó que només s'envia la informació que ha canviat (amb informació de control, òbviament). Per tant, el volum transmès és menor que si s'enviés l'arxiu sencer.
- kioslave: sistema part de l'entorn gràfic KDE per gestionar els accessos a disc.

j) Seguretat en la transmissió:

- Depèn: en els casos en els que es permeten diferents protocols per transmetre la informació, en funció del sistema que s'utilitzi hi haurà seguretat o no. Per exemple, un mateix programari pot utilitzar *ssh*, és a dir, hi haurà seguretat intrínseca a la connexió o pot utilitzar *ftp*, i per tant, les dades viatjaran en pla. També es pot donar el cas en que les dades viatgin en pla, però xifrades en origen, per exemple, utilitzant *gnupg*.
- SSL: el canal s'estableix utilitzant el protocol Secure Socket Layer, de manera que garanteix la confidencialitat de la transmissió.
- GnuPG: els arxius que s'envien han sigut xifrats en origen. Això vol dir, que la informació no viatjarà en pla, però només es refereix als arxius, i no pas al canal.

k) Cost econòmic:

- 0€: s'entén que el programari no té cap cost d'adquisició, llicència o utilització. Això no treu que hi hagin costos de personal, maquinari, connexió a Internet...

l) Personal necessari

- Usuari: entès com que una persona que no té formació tècnica, és a dir, que utilitza habitualment l'ordinador, però no té coneixements específics. El que popularment es coneix com un "nivell d'usuari".
- Tècnic: que seria una persona que té coneixements informàtics avançats, que es podrien concretar en la instal·lació, configuració i manteniment de sistemes operatius, així com domini de la línia de comandes.

m) Control de canvis:

S'entén que control de canvis es refereix a com el programari gestiona què ha canviat dins de l'arxiu.

- Data, mida: s'utilitza la informació del sistema de fitxers sobre l'arxiu per determinar si ha canviat. Es pot donar, però que un arxiu hagi canviat.
- Gnome: s'utilitza la infraestructura d'aquest entorn gràfic per realitzar les operacions de còpia.
- MD5: utilitza aquest algorisme per calcular el resum de cada arxiu, i fer la comparació. D'aquesta manera només ha de contrastar els 128bits de resum de cada arxiu.
- Nautilus: s'utilitza la infraestructura d'aquest programari de gestió d'arxius de l'entorn gràfic Gnome.
- Rdiffdir: procés per generar delta respecte d'un directori.
- Rsync: tècnica que utilitza l'algorisme per, a partir dels *checksums* dels dos arxius, generar, informar dels canvis que hi ha hagut.

n) Direccionalitat

Unidireccional: aquests programaris només permeten fer la sincronització en una sola

direcció, de manera que, en alguns casos podrien donar resposta, però requereixen una atenció prèvia de l'usuari en cas que faci canvis en ambdós costats.

Aquí hi ha programaris que han estat concebuts com a còpia de seguretat, però que pel fet de deixar una rèplica operativa en l'altre costat, permeten fer la funció de sincronització. També, al tractar un tema molt semblant però des d'una òptica diferent, aporten idees de com resoldre alguns problemes.

Els que estan dissenyats d'aquesta manera, per tant, no se'ls hi apliquen les comprovacions sobre els canvis en un costat, en l'altre, en els dos, en els dos amb els mateixos arxius, amb els canvis de nom, amb els moviments ni amb la detecció de conflictes per haver editat en ambdós costats. Això és perquè els programes assumeixen l'altre costat com un espai on deixaran una còpia que no serà modificada.

Bidireccional: programaris que treballen en ambdues direccions i que per tant ens permetrien fer canvis en qualsevol dels costats sense estar pendents d'aquest fet. Aquests serien els que responen més a les necessitats plantejades.

o) Gestió d'arxius:

- Correcte: ha passat satisfactòriament la prova sense necessitat d'intervenció de l'usuari.
- Conflicte: la prova ha finalitzat correctament però requereix de la intervenció de l'usuari. Veure l'apartat "Conflictes" de la fitxa per veure com es resol.

p) Conflictes:

- Preguntar: sol·licita la intervenció de l'usuari per resoldre la situació.

4.3. Programaris aptes

Per cadascun dels programaris que s'han provat, hi ha creada una fitxa amb els resultats, així com la informació extra obtinguda. Com s'ha comentat en el punt anterior, en funció de si estan dissenyats per treballar bidireccionalment tindran resultats de proves que no són aplicables als que ho fan unidireccionalment.

Per fer la fitxa s'han reorganitzat els ítems i les proves, agrupant-les per conceptes.

a) BackInTime

- Programari
 - Versió 0.9.26-4
 - Llicència GPL
 - Automatització Si
 - Portabilitat No
 - Dependències Python
 - Interfície Gràfica
 - Estructura Client
 - Emmagatzematge suportat Qualsevol (amb enllaços forts)
- Transmissió
 - Protocol en la transmissió rsync
 - Compensió en la transmissió rsync
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis rsync
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - La còpia pot funcionar en segon pla, llençant les comandes per consola.
 - Com que utilitza enllaços forts, a cada directori de còpia, es pot accedir a una imatge sencera de l'original, tenint en compte que cada arxiu només hi és un cop al disc.
 - Per tant, la còpia a "l'altre banda" és totalment utilitzable.
 - No treballa directament a través de la xarxa es pot fer, per exemple, muntant el volum remot.
- Web <http://backintime.le-web.org/>

b) backup-manager

- Programari
 - Versió 0.7.9-3
 - Llicència GPL
 - Automatització No directa
 - Portabilitat No
 - Dependències
 - Interfície Consola
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió ftp, ssh, rsync, Amazon S3
 - Compensió en la transmissió No (excepte rsync)
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis Data, mida
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - La còpia a l'altre costat queda desada en arxius comprimits amb formats oberts.
 - Hi ha la opció d'utilitzar els serveis d'emmagatzematge d'Amazon com a còpia externa.
 - Possibilitat de xifrar els arxius.
- Web <http://www.backup-manager.org/>

c) backuppc

- Programari
 - Versió 3.1.0-9
 - Llicència GPL
 - Automatització Si
 - Portabilitat Windows, Mac
 - Dependències apache, perl, samba, bzip2, tar
 - Interfície Web
 - Estructura Client – Servidor
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió Samba, rsync (via ssh, rsh), nfs, ftp
 - Compensió en la transmissió No (excepte rsync i bzip2)
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis MD5
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - No guarda dos cops el mateix arxiu, manté una relació dels arxius modificats.
 - El fet de gestionar-se a través de la web, fa que es pugui controlar independentment del lloc o la plataforma.
- Web <http://backuppc.sourceforge.net/>

d) Conduit

- Programari
 - Versió 0.3.17-1
 - Llicència GPL
 - Automatització Si
 - Portabilitat Gnome
 - Dependències Python, Gtk
 - Interfície Gràfica
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió Gnome
 - Compensió en la transmissió No
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis Gnome
 - Reanomenar Copia l'arxiu "nou" a l'altre costat i considera un conflicte l'antic
 - Moviment Ídem que l'anterior
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Correcte
 - Modificació d'arxius diferents en dos costats Correcte
 - Modificació dels mateixos arxius en els dos costats Conflicte
 - Conflictes Es resolen manualment des del programa. El programari avisa canviant la icona a la safata de l'entorn gràfic.
- Observacions

- Inclou el mode “always up to date”, tot i que les sincronitzacions s'han d'activar manualment.
 - Pel que fa a la sincronització d'arxius, no té connector per xarxa, és a dir, que no permet connectar amb una font externa a la màquina. Es pot solucionar (provat) muntant el directori remot, per exemple amb NFS.
 - Projecte en una fase molt inicial, versió estable amb errades notables, inestable.
 - Facilitat per sincronitzar amb serveis com Google Docs, Flickr, Facebook...
 - La gestió dels conflictes no és clara, sobretot si el camí cap a l'arxiu és llarg. En algunes circumstàncies només permet esborrar o mantenir l'arxiu conflictiu.
- Web <http://live.gnome.org/Conduit/>

e) crsync2

- Programari
 - Versió 1.34-2.2
 - Llicència GPL
 - Automatització No directa
 - Portabilitat No
 - Dependències rsync, gnutls, sqlite
 - Interfície consola
 - Estructura Client – Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió rsync
 - Compensió en la transmissió rsync
 - Seguretat en la transmissió SSL
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis rsync
 - Reanomenar Correcte
 - Moviment Correcte
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Correcte
 - Modificació d'arxius diferents en dos costats Correcte
 - Modificació dels mateixos arxius en els dos costats Correcte
 - Conflictes Preguntar
- Observacions
 - Orientat a la gestió de les configuracions dels clústers.
 - Configuració complexa.

- La informació de l'estat dels arxius es desa en una base de dades en cada hoste. D'aquesta manera gestiona de manera correcta els moviments, els canvis de nom, els arxius esborrats o les modificacions en paral·lel.
- Pot llençar comandes després de fer la sincronització.
- Web <http://oss.linbit.com/csync2/>

f) Dropbox

- Programari
 - Versió 0.6.7
 - Llicència GPL (client)
 - Automatització Si
 - Portabilitat Windows, Mac, Android, iPhone i Blackberry
 - Dependències Nautilus
 - Interfície Gràfica
 - Estructura Client – Servidor
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió Delta encoding
 - Compressió en la transmissió Delta encoding
 - Seguretat en la transmissió SSL
- Recursos necessaris
 - Cost econòmic, 0€ (en modalitat bàsica de 2GB d'emmagatzematge)
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis Nautilus
 - Reanomenar Correcte
 - Moviment Correcte
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Correcte
 - Modificació d'arxius diferents en dos costats Correcte
 - Modificació dels mateixos arxius en els dos costats Conflicte
 - Conflictes Es creen dos arxius, el més “nou” se li afegeix al nom la paraula “conflict” junt amb la marca de temps de quan s'ha creat.
- Observacions

- Necessita del servidor intermediari al núvol.
- Permet tenir tants clients funcionant alhora com sigui necessari.
- Concepte interessant.
- És transparent i no necessita d'intervenció de l'usuari.
- Historial de canvis dels arxius.
- Es pot accedir als arxius, així com a l'historial a través de la web del programari.
- A l'haver-hi (com a mínim) una tercera còpia que a més guarda versions antigues, també és un sistema de còpia de seguretat.
- En cas d'interrupció de la connexió, sincronitza quan aquesta es recupera.
- Web <http://www.dropbox.com>

g) duplicity

- Programari
 - Versió 0.6.10
 - Llicència GPL
 - Automatització No directa
 - Portabilitat No
 - Dependències Rsync, gnupg, python
 - Interfície Consola
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió ftp, ssh, rsync, Amazon S3
 - Compressió en la transmissió No (excepte rsync)
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis rdiffdir
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - Un altre exemple de com recombinant altres programaris ja existents de reputació contrastada, es pot crear un programari eficient.
 - La còpia a l'altre costat queda desada en arxius comprimits amb formats oberts.
 - Hi ha la opció d'utilitzar els serveis d'emmagatzematge d'Amazon com a còpia externa.
 - Possibilitat de xifrar els arxius.
- Web <http://duplicity.nongnu.org/>

h) grsync

- Programari
 - Versió 1.1.1-1
 - Llicència GPL
 - Automatització No
 - Portabilitat Windows, Mac
 - Dependències Gtk, rsync
 - Interfície Gràfica
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió Rsync, rsh, ssh
 - Compensió en la transmissió rsync
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis rsync
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - És una bona manera d'utilitzar el programari *rsync* d'una manera senzilla, doncs a través de la interfície gràfica es genera la comanda corresponent.
 - Dit d'una altra manera, fa accessible totes les possibilitats de sincronització als usuaris que no tenen un perfil tècnic.
- Web <http://www.opbyte.it/grsync/>

i) Krusader

- Programari
 - Versió 2.2.0-beta1
 - Llicència GPL
 - Automatització No
 - Portabilitat KDE
 - Dependències KDE
 - Interfície Gràfica
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió kioslave
 - Compensió en la transmissió kioslave
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis MD5
 - Reanomenar Correcte
 - Moviment Correcte
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Correcte
 - Modificació d'arxius diferents en dos costats Correcte
 - Modificació dels mateixos arxius en els dos costats Falla, l'arxiu més “nou” és el que es replica
 - Conflictes Preguntar
- Observacions
 - Els moviments i els canvis de nom els considera com un esborrat i la creació d'un arxiu nou.

- Molt clar a nivell visual, queda evident que s'està treballant amb dos directoris.
 - Permet fer altres operacions de gestió amb els arxius (per exemple, copiar, moure, descomprimir...) a part d'incloure complements per poder fer altres operacions no tant trivials com mesclar arxius, comparar-los de manera visual en quan a contingut...
 - Els accessos a arxius remots estan condicionats totalment a la connexió, ja que el tractament és el mateix que als locals.
- Web <http://www.krusader.org/>

j) inosync

- Programari
 - Versió 0.2.1-1
 - Llicència GPL
 - Automatització Si
 - Portabilitat No
 - Dependències Python, rsync
 - Interfície Consola
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió rsync
 - Compensió en la transmissió rsync
 - Seguretat en la transmissió Rsync, rsh, ssh
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis rsync
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - Els canvis no s'han d'anunciar, vigila els canvis en el sistema d'arxius, d'aquesta manera, sempre que algun arxiu es modifica o esborra o es crea, és automàticament sincronitzat.
 - Aporta el fet de no haver de programar les sincronitzacions, sinó d'estar pendent dels canvis.
 - Poc documentat.
- Web <https://github.com/hollow/inosync>

k) syrep

- Programari
 - Versió 0.9-4
 - Llicència GPL
 - Automatització No directa
 - Portabilitat Windows
 - Dependències
 - Interfície Consola
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió Sistema
 - Compensió en la transmissió No
 - Seguretat en la transmissió No
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis MD5
 - Reanomenar Correcte
 - Moviment Correcte
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Correcte
 - Modificació d'arxius diferents en dos costats Correcte
 - Modificació dels mateixos arxius en els dos costats Conflicte
 - Conflictes Preguntar
- Observacions
 - Projecte abandonat, no es manté ni s'actualitza des del 2006.
 - Manté una llista de les transferències, canvis que hi ha hagut, d'aquesta manera

detecta els moviments, els esborrats...

- És capaç de generar pedaços respecte a un altre còpia, de manera que existeix la possibilitat d'enviar o portar un arxiu amb els canvis, sense tenir connexió directa entre les dues parts.

- Web

<http://0pointer.de/lennart/projects/syrep/>

I) **rsnapshot**

- Programari
 - Versió 1.3.1-1
 - Llicència GPL
 - Automatització No directa
 - Portabilitat Mac
 - Dependències Perl, rsync
 - Interfície Consola
 - Estructura Client
 - Emmagatzematge suportat Qualsevol (amb enllaços forts)
- Transmissió
 - Protocol en la transmissió Rsync, ssh
 - Compressió en la transmissió rsync
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis rsync
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - Utilitza els enllaços forts.
 - A l'altre costat totes les còpies tenen tots els arxius (tot i que en realitat al disc, cadascun només hi és una sola vegada), i per tant, són totalment funcionals.
 - Un altre exmple de com la recombinació de programaris ja existents i contrastats poden generar propostes molt interessants.
- Web <http://www.rsnapshot.org/>

m) rsync

- Programari
 - Versió 3.0.7-2
 - Llicència GPL
 - Automatització No directa
 - Portabilitat Windows
 - Dependències
 - Interfície Consola
 - Estructura Client, Client – Client, Client – Servidor
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió rsync, rsh, ssh
 - Compensió en la transmissió rsync
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis rsync
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - És el programari base de molts altres, degut a la seva eficiència.
 - Procedeix d'un bon model teòric, d'una implementació molt eficient de la sincronització només dels canvis.
 - Quan treballa en remot, necessita del programari servidor funcionant a l'altre costat o de poder executar-hi comandes.
 - Requereix molt de càlcul en favor de minimitzar la taxa de transferència.
- Web <http://rsync.samba.org/>

n) ubuntuone-client

- Programari
 - Versió 1.4.4.1-0
 - Llicència GPL (client)
 - Automatització Si
 - Portabilitat Windows, Android
 - Dependències Python, Nautilus, Gnome
 - Interfície Gràfica
 - Estructura Client – Servidor
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió u1sync
 - Compensió en la transmissió No
 - Seguretat en la transmissió SSL
- Recursos necessaris
 - Cost econòmic, 0€ (en modalitat bàsica de 2GB d'emmagatzematge)
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis Nautilus
 - Reanomenar Correcte
 - Moviment Correcte
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Correcte
 - Modificació d'arxius diferents en dos costats Correcte
 - Modificació dels mateixos arxius en els dos costats Falla, el primer que actualitza, “guanya”
 - Conflictes A través de comandes
- Observacions
 - Necessita del servidor intermediari al núvol.
 - Concepte interessant.

- No està pensat perquè es facin canvis en dos llocs alhora, es pressuposa un sol usuari que està treballant en un sol lloc a la vegada.
- És molt senzill d'utilitzar, transparent i pràcticament no necessita d'intervenció de l'usuari.
- En cas d'interrupció de la connexió, sincronitza quan aquesta es recupera.
- Web <http://one.ubuntu.com>

o) unison-gtk

- Programari
 - Versió 2.32.52-1
 - Llicència GPL
 - Automatització No
 - Portabilitat Windows
 - Dependències Gtk
 - Interfície Gràfica
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió rsync
 - Compensació en la transmissió rsync
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Usuari
- Gestió dels arxius
 - Control de canvis rsync
 - Reanomenar Correcte
 - Moviment Correcte
 - Direccionalitat Bidireccional
 - Primera còpia Correcte
 - Modificació d'arxius en un costat Conflicte
 - Modificació d'arxius diferents en dos costats Conflicte
 - Modificació dels mateixos arxius en els dos costats Conflicte
 - Conflictes Preguntar
- Observacions
 - Basat en un gran model teòric.
 - Té una versió només per consola, que es pot automatitzar de manera “No directa”.

- Manté un historial de l'estat de cada arxiu, d'aquesta manera pot gestionar les modificacions, els esborraments...
- Està pensat per treballar només entre dues màquines, tot i que, es pot treballar amb més si es fa servir una topologia d'estrella en la que una fa de servidor.
- Web <http://www.cis.upenn.edu/~bcpierce/unison/>

p) zsync

- Programari
 - Versió 0.6.1-1
 - Llicència Artistic License
 - Automatització No directa
 - Portabilitat No
 - Dependències
 - Interfície Consola
 - Estructura Client
 - Emmagatzematge suportat Qualsevol
- Transmissió
 - Protocol en la transmissió rsync
 - Compressió en la transmissió rsync
 - Seguretat en la transmissió Depèn
- Recursos necessaris
 - Cost econòmic, 0€
 - Personal necessari Tècnic
- Gestió dels arxius
 - Control de canvis Arxiu amb els canvis
 - Direccionalitat Unidireccional
 - Primera còpia Correcte
- Observacions
 - Utilitza la teoria del *delta encoding* i del *rsync* per descarregar arxius: a partir de l'arxiu en local, i d'un arxiu intermedi amb els canvis que s'han produït, només obtindrà el que hagi canviat, minimitzant el volum de descàrrega.
 - En una descàrrega web, per exemple, necessita que s'hagi generat l'arxiu que conté la informació dels canvis.
- Web <http://zsync.moria.org.uk/>

4.4. Programaris descartats

De la llista inicial, n'hi ha alguns que han caigut, ja sigui perquè des d'un primer moment s'ha vist que no podien complir la funció de sincronització, ja sigui perquè al fer les proves s'ha vist que eren molt semblants algun ja provat, i que no aportaven res de nou. Així doncs, la llista és la següent:

Programari	Motiu d'exclusió
amanda	Programari clarament orientat a la còpia de seguretat, que planteja una infraestructura molt complexa, i alhora no permet treballar de manera senzilla amb la còpia.
dirvish	Projecte abandonat, molt semblant al <i>backuppc</i>
egnyte	Mateixes característiques que el programari <i>dropbox</i> .
flamethrower	Component del programari <i>SystemImager</i> , que té l'objectiu de facilitar les instal·lacions en massa.
flickrfs	Programari orientat per muntar l'espai de dades de flickr en un directori local. Només està dissenyat per aquest servei.
gerstensaft	Versió gràfica del projecte <i>sendfile</i> , projecte abandonat i obsolet.
komparator	Projecte abandonat. Funcionalitat igual que el programari <i>krusader</i>
mirror	Projecte de sincronització obsolet, només comprova els canvis utilitzant la data/mida de l'arxiu, així com només permet fer transferències FTP.
opensync-plugin-file	Part d'un projecte molt ambiciós, però encara en una fase molt inicial.
rdiff-backup	Veure <i>rsnapshot</i> .
rdiff	Part essencial del programari <i>rsync</i> per calcular els <i>delta</i> , però no per realitzar les tasques de sincronització.
rdup	Programari per crear llistes de arxius que han canviat.
sendfile	Interessant concepte d'intercanvi d'arxius utilitzant un servidor FTP intermedi, que ha quedat obsolet amb l'arribada del núvol.
simba	Projecte abandonat.
subversion-tools	Conjunt d'eines que acompanyen el programari de control de versions <i>subversion</i> .
wuala	Mateixes característiques que el programari <i>dropbox</i> .
zumodrive	Mateixes característiques que el programari <i>dropbox</i> .

5. Conclusions

5.1. Quines característiques hauria de tenir un programari de sincronització?

A partir de les informacions teòriques i de les recollides en les proves, sorgeixen les característiques més interessants, en el benentès que facilitaran la sincronització d'arxius, és a dir, mantenir dues còpies exactes d'un conjunt d'arxius en llocs diferents, evitant els conflictes, minimitzant el temps i els recursos consumits:

Mantenir una base de dades amb l'estat dels arxius en cada moment, tant dels que estan en local com els que estan en remot, per poder estar al cas de les modificacions, sobretot pel que fa al cas en el que el mateix arxiu es modifica en diferents llocs en el mateix moment, tot evitant la eliminació involuntària d'una part de la feina feta.

Si aquesta base de dades és capaç de mantenir informació sobre el contingut, o rep notificacions per part del sistema de fitxers, també notificarà els moviments en directoris o els canvis de nom. D'aquesta manera, l'actualització en l'altre costat serà immediata. Amb aquesta manera de procedir, com a contraexemple, un arxiu de 4GB canviat de directori, no representarà cap maldecap.

Capacitat de generar pedaços respecte a un altre còpia, de manera que existeixi la possibilitat d'enviar o portar un arxiu amb els canvis, sense tenir connexió directa entre les dues parts, o enviar aquest en el moment de connexió.

Claredat a nivell visual, que quedi evident que s'està treballant amb dos directoris diferents i senzillesa per resoldre els conflictes, i mínima necessitat d'intervenció per part de l'usuari.

Facilitat per sincronitzar amb serveis al núvol com Google Docs, Flickr, Facebook, Amazon... en aquest sentit l'avantatge de treballar amb el núvol és que no es sobrecarrega un servidor, hi ha un intermediari que fa la feina bruta. Enviar-ho al núvol, a més, et dóna una còpia de seguretat extra. Si a més, el sistema preveu un accés als arxius via web, facilita una altra manera d'accedir a la informació.

Poder llençar comandes després de fer la sincronització.

En cas d'interrupció de la connexió, sincronitzar automàticament quan aquesta es

recupera.

Permetre tenir tants clients funcionant alhora com sigui necessari.

Bidireccionalitat per defecte, no haver de gestionar en quin costat s'han fet els canvis.

Per seguretat, possibilitat de xifrar els arxius.

No haver d'anunciar els canvis, que els vigili respecte el sistema d'arxius, d'aquesta manera, sempre que algun arxiu es modifica o esborra o es crea, és automàticament sincronitzat.

Enviament només dels canvis que s'han produït en l'arxiu, en comptes del contingut sencer. En aquest sentit, exemples com el *rsync*, és molt interessant com resol el fet de enviar només els canvis quan potser només ha canviat una part que a més s'ha desplaçat en l'arxiu; requereix molt de càlcul en favor de minimitzar la taxa de transferència.

5.2. Proposta

A partir de la informació recollida sobre els programaris, de la selecció de les millors característiques del punt anterior, i cada cas en particular, les propostes són les següents:

a) Cas número 1

Donada la tipologia dels usuaris, la diversificació de sistemes operatius, Dropbox per la senzillesa, pels costos d'instal·lació (a part de la infraestructura necessària, només caldria pagar per la quantitat de dades emmagatzemades) i per la transparència de tot el procés.

Però, aquí també entra en joc un tema aliè a la senzillesa, ja que aquestes persones tracten amb informació sensible, entra en joc la protecció de dades, i els dubtes que pot generar, respecte de com s'emmagatzemen, fet que pot portar a F a decidir no optar per tenir dades en un lloc que desconeixen.

En aquest cas, Conduit per senzillesa, tot i que seria necessari esperar a una versió més estable, i que requeriria que l'accés que tenen a la informació permeti treballa amb volums com si estiguessin en local.

Sinó, Krusader per la possibilitat d'instal·lar-lo en altres plataformes, junt amb la senzillesa visual, que tot i que no és automàtic, a l'usuari li queda molt clar que està sincronitzant canvis.

b) Cas número 2

Com que es tracta d'un perfil tècnic, la opció més òptima seria Unison utilitzant una connexió via SSH, donat que és possible que en totes les màquines hi hagi instal·lat per defecte el programari *rsync* i que ja hi hagi un accés SSH configurat per l'usuari, tant per enviar/rebre els canvis, com per utilitzar comandes per parar serveis, configurar...

c) Cas número 3

En aquest cas es pot descartar el treball amb el núvol, donat que un projecte de 20GB que s'hagi de sincronitzar per primer cop, trigarà (assumint que la velocitat de pujada mitjana de les connexions al país està en 500kbps), al voltant de 4 dies sencers.

Per tant, la proposta seria la d'utilitzar *grsync* (que està portat a MacOS), més tenint en compte que és un sol ordinador el que està en joc amb les seves memòries externes, i per tant, és senzill de moure la informació d'un costat a l'altre.

d) Cas número 4

Tenint en compte que els arxius han d'estar a tot arreu, junt amb la possibilitat d'usar-los tant des de Windows com des de Linux, Dropbox és una opció molt interessant, més amb la opció de compartir espai amb altres usuaris, fet que permetria tenir a cada persona el seu espai, junt amb una part comuna.

Com que hi hauran uns costos associats al volum de dades emmagatzemades en remot, i que probablement, hi haurà més dades públiques, i que hi haurà servidors connectats sempre a Internet, és interessant la opció de Unison (utilitzant la topologia d'estrella recomanada en el manual) aprofitant la infraestructura pròpia. D'aquesta manera, els costos o el temps de gestió pot estar diluït entre les funcions pròpies dels tècnics.

5.3. Propostes de millora

Com a possibilitat de desenvolupament, donat que és un programari robust i de codi lliure, afegir a Unison la capacitat d'estar funcionant en tot moment i de detectar els canvis en el sistema de fitxers.

Per altra banda, no estaria de més millorar la interfície gràfica per fer-la més accessible a perfils no tècnics

5.4. Recomanacions

Paral·lelament a les conclusions i propostes generades, sorgeix un llistat de recomanacions per tenir còpies distribuïdes. El fet de tenir dades que viatgen, en màquines que són fàcilment portables, sense oblidar que aquestes dades també es mouen per la xarxa, fa que sigui aconsellable:

- particions xifrades: que la informació que està guardada en màquines que estan repartides pel món no estigui desada en pla, perquè en el cas d'accés no autoritzat o robatori, no sigui visible.
- contrasenyes fortes: que l'accés a les màquines, al sistema operatiu com a les aplicacions que sincronitzen les dades, tinguin contrasenyes que no siguin fàcilment deduïbles o trencables.
- transferència segura: sempre utilitzar canals que xifren les dades mentre viatgen per la xarxa, per assegurar-se que qui les rep és el receptor escollit.
- canals només orientats a la transferència: vigilar quan s'utilitzen canals que es poden fer servir per altres usos, per exemple, al donar accés estàndard per SSH, també es permet executar comandes aleatòries a la màquina a la qual s'accedeix.
- quins arxius es sincronitzen: llistar els arxius o directoris que realment seran necessaris, per no abocar a la xarxa informació que no ha de ser consultada, i que per tant, no té motiu de ser que estigui físicament fora de l'empresa.

6. Annexes

6.1. Modificació automàtica d'arxius

Per seleccionar els arxius que es modificaran, programat en *bash* (per la senzillesa en gestionar arxius), que a partir del llistat d'arxius disponibles al directori “repositori”, en seleccionen un subconjunt aleatòriament, i el passa a uns altres *scripts* en funció de les extensions:

```
#!/bin/bash

APP=$1

if [ -z $APP ]
then
    echo "Utilització: $0 <test_app>"
else
    DIR="repositori/"
    EXTENSIONS="png odt"
    HOSTNAME=`hostname -s`
    TIMESTAMP=`date +%Y%m%d-%H%M%S`
    LOGDIR="log/"

    for EXT in $EXTENSIONS
    do
        FILENAME=$HOSTNAME-$TIMESTAMP-$EXT
        FILELIST=$LOGDIR/$APP-filelist-$FILENAME.log
        MODIFLIST=$LOGDIR/$APP-modiflist-$FILENAME.log
        find $DIR -name \*.$EXT >> $FILELIST
        NUMFILES=`cat $FILELIST | wc -l`
        let "NUMMODIF=($RANDOM % $NUMFILES) +1"
        echo "Es modificaran $NUMMODIF arxius $EXT"
        for ((I=1; I<=$NUMMODIF; I++))
        do
            let "MODIFNUM=($RANDOM % $NUMMODIF) +1"
            MODIFILE=`sed -n "$MODIFNUM"p $FILELIST`
            echo $MODIFILE >> $MODIFLIST
            echo "Modificant: $MODIFILE"
            MODIFILES=`echo $MODIFILE | sed -e 's/ /\ \ /g'`
            ./modif-$EXT $MODIFILES
        done
    done
done
fi
```

Nota: tot i que no ha afectat al funcionament de les proves, al seleccionar-se aleatòriament els arxius, en la prova de les modificacions en ambdós costat en arxius diferents, s'ha donat ocasionalment que es modificava el mateix arxiu.

Una versió del *script* anterior és el que permet, a partir d'una llista d'arxius modificats aplicar les modificacions, és a dir, no els selecciona aleatòriament de la llista de tots els arxius, sinó que els extreu d'una llista:

```
#!/bin/bash

APP=$1
FILELIST=$2

if
  [ -z $APP ]
  [ -z $FILELIST ]
then
  echo "Utilització: $0 <test_app> <file_list>"
else
  DIR="repositori/"
  EXTENSIONS="png odt"
  HOSTNAME=`hostname -s`
  TIMESTAMP=`date +%Y%m%d-%H%M%S`
  LOGDIR="log/"

  for EXT in $EXTENSIONS
  do
    FILENAME=$HOSTNAME-$TIMESTAMP-$EXT
    MODIFLIST=$LOGDIR/$APP-modiflist-$FILENAME.log
    NUMMODIF=`cat $FILELIST$EXT.log | wc -l`
    echo "Es modificaran $NUMMODIF arxius $EXT"
    for ((I=1; I<=$NUMMODIF; I++))
    do
      MODIFILE=`sed -n "$I"p $FILELIST$EXT.log`
      echo $MODIFILE >> $MODIFLIST
      echo "Modificant: $MODIFILE"
      MODIFILES=`echo $MODIFILE | sed -e 's/ /\ \ /g'`
      ./modif-$EXT $MODIFILES
    done
  done
fi
```

D'aquesta manera, es pot fer la prova per quan es modifica el mateix arxiu en les dues bandes.

Els arxius que es modificaran per la simulació de la naturalitat del dia a dia, seran documents de text i imatges. El llenguatge utilitzat és *Perl* per la facilitat que, amb les extensions, permet treballar els continguts dels arxius

Per modificar els arxius *odt*, és a dir, documents de text, el següent *script* selecciona una paràgraf a l'atzar i el subratlla, com si d'una revisió es tractés:

```
#!/usr/bin/perl -w

use strict;
use OpenOffice::OODoc;

if (@ARGV) {

    my ($doc, @elements, $modifyElement);
    $doc = odfDocument(file => $ARGV[0]);
    $doc->createStyle ( "Subrallat", family => 'paragraph',
parent => 'Standard', properties => { 'fo:background-color' =>
'#ffff00' } );
    @elements = $doc->getTextElementList();
    $modifyElement = rand(scalar @elements);
    $doc->textStyle($elements[$modifyElement], "Subrallat");
    $doc->save();

}
else {
    print "Utilització: perl modif-odt.pl <doc.odt>\n";
}
```

Per modificar els arxius *png*, és a dir, imatges, el següent *script* pinta 20 punts de color aleatòriament a la imatge:

```
#!/usr/bin/perl -w

use strict;
use Image::Magick;

if (@ARGV) {

    my ($image, $width, $height, $i, $file);

    $file = $ARGV[0];
    $image = new Image::Magick;
    $image->Read($file);
    $width = $image->Get('columns');
    $height = $image->Get('rows');
    for ($i = 0; $i < 20; $i++) {
        $image->Draw(fill =>
'rgb(' . rand(255) . ',' . rand(255) . ',' . rand(255) . ')',
primitive=>'point', points=>rand($width) . ',' . rand($height));
    }
    $image->Write($file);
}
else {
    print "Utilització: perl modif-png.pl <img.png>\n";
}
```

6.2. Comprovació automàtica dels canvis

Per comprovar els canvis que hi ha hagut en els arxius després de les proves, és a dir, quins estan sincronitzats i quins no, la tècnica (també programada en *bash*) és, en ambdues bandes, la de llistar tots els arxius, calcular la suma MD5 i seguidament comparar-les. Si no retorna res, és que ambdues parts estan sincronitzades, sinó mostrarà, quins arxius estan només en un costat i quins són diferents.

```
#!/bin/bash

APP=$1

DIRL="repositori/"
DIRR="/mnt/shared/repositori/"
LOGDIR="/mnt/shared/log/"

if [ -z $APP ]
then
    echo "Utilització: $0 <test_app>"
else
    HOSTNAME=`hostname -s`
    TIMESTAMP=`date +%Y%m%d-%H%M%S`
    FILENAMEL=$LOGDIR/$APP-check-L-$HOSTNAME-$TIMESTAMP.log
    FILENAMER=$LOGDIR/$APP-check-R-$HOSTNAME-$TIMESTAMP.log
    LISTL=`find $DIRL -type f`
    LISTR=`find $DIRR -type f`

    for FILE in $LISTL
    do
        md5sum $FILE >> $FILENAMEL
    done
    for FILE in $LISTR
    do
        md5sum $FILE >> $FILENAMER
    done
    diff $FILENAMER $FILENAMEL
fi
```

7. Bibliografia

- Pierce, Benjamin C. et al., *What is a File Synchronizer – Technical report* (1998)
<http://www.cis.upenn.edu/~bcpierce/papers/snc.ps>
- Tridgell, Andrew, *Efficient Algorithms for Sorting and Synchronization* (1999)
http://samba.org/~tridge/phd_thesis.pdf

Amazon S3	http://aws.amazon.com/s3/
apt	http://www.debian.org/doc/manuals/apt-howto/
Artistic License	http://www.opensource.org/licenses/artistic-license.php
bash	http://www.gnu.org/software/bash/manual/bashref.html
cron	https://help.ubuntu.com/community/CronHowto
Cygwin	http://www.cygwin.com/docs.html
Delta encoding	http://www.dspguide.com/ch27/4.htm
GNU GPL	http://www.gnu.org/licenses/gpl.html
GnuPG	http://www.gnupg.org/
kioslave	http://api.kde.org/4.x-api/kdelibs-apidocs/kio/html/index.html
Kubuntu 10.04	http://www.kubuntu.org/
Nautilus	http://live.gnome.org/Nautilus
NFS (Network File System)	https://help.ubuntu.com/community/SettingUpNFSToHowTo
Perl	http://perldoc.perl.org/
Perl – CPAN	http://www.cpan.org/
Python	http://www.python.org/
rdiffdir	http://www.nongnu.org/duplicity/rdiffdir.1.html
rsync	http://rsync.samba.org/documentation.html
SSH	http://www.openssh.com/manual.html
Ubuntu 10.10	http://www.ubuntu.com
vmware Player 3.01	http://www.vmware.com/products/player/